Head Pose Estimation Based on Face Symmetry Analysis

Afifa Dahmane, Slimane Larabi, Ioan Marius Bilasco, Chaabane Djeraba

Noname manuscript No.
(will be inserted by the editor)

Head Pose Estimation Based on Face Symmetry Analysis

Afifa Dahmane · Slimane Larabi · Ioan Marius Bilasco · Chabane Djeraba

Received: date / Accepted: date

Abstract This paper address the problem of head pose estimation in order to infer a non-intrusive feedback from users about gaze attention. The proposed approach exploit the bilateral symmetry of the face. We use the size and the orientation of the symmetrical area of the face to estimate roll and yaw poses by the mean of Decision Tree model. The approach does not need the location of interest points on face and is robust to partial occlusions. Tests were performed on different datasets (FacePix, CMU PIE, Boston University) which give variability in illumination and expressions. Results demonstrate that the change in the size of the regions that contain a bilateral symmetry provides accurate pose estimation.

Keywords Head pose estimation · Symmetry detection · Pattern recognition

1 Introduction

The head pose is often linked with visual gaze estimation and provides a coarse indication of the gaze in situations where the system should be non-intrusive using only a regular camera either in situations where the eyes may be not visible. In this context, a coarse head pose can give a good indication of the gaze attention.

Afifa Dahmane, Slimane Larabi
Computer Science Department, USTHB University, Algeria
Tel.: +213 21247607
Fax: +213 21247607
E-mail: fdahmane,slarabi@usthb.dz

Afifa Dahmane, Ioan Marius Bilasco, Chabane Djeraba
LIFL, USTL Universitiy of Lille UMR CNRS 8022, France
Tel.: +33 3 62531584
Fax: +33 3 28778537
E-mail: afifa.dahmane,marius.bilasco,chabane.djeraba@lifl.fr

Head pose estimation is a classic problem in computer vision. It is widely used in many applications such as video conferencing, driver monitoring or human computer interaction. Moreover, for many pattern recognition applications, it is necessary to estimate coarse head pose to eliminate variation in pose for better accuracy (e.g. face recognition or facial expression analysis). Many approaches based on local facial features are proposed to deal with head pose estimation. However, the obvious difficulty for this local approaches lies in detecting outlying or missing features in situations where facial landmarks are obscured. Also, low resolution imagery make it difficult to precisely determine the feature locations.

1.1 Contribution

This paper presents a method based on symmetry to estimate discrete head pose. We exploit the bilateral symmetry of the face to directly deduce two degrees of freedom for the head (yaw and roll). The symmetry is defined using global skin region instead of local interest points. The proposed approach does not need the location of interest points on the face and can be deployed using low-cost and widely available hardware. Also, no initialization of pose nor calibration are required. The estimated pose is coarse but sufficient to infer general gaze direction. We have three main contributions:

- First, we develop a method for detecting the position of symmetry axis and its orientation in an image.
- Second, the roll angle is deduced from the inclination of the symmetry axis.
- Third, the yaw angle is calculated using the region which delimits symmetrical pixels.

Symmetrical region is defined by analysing pixels intensity. The intensity of one pixel on the right side of the face is more similar to its mirror pixel than another pixel in the image. We have conducted experiments which indicate that the use of facial symmetry as a geometrical indicator for head pose is still reliable when local geometric features (such as eyes, nose or mouth) are missed due to occlusions or wrong detections. We give more insights about the method comparatively with our previous work and extend it to two degrees of freedom. Besides using public datasets (FacePix, CMU PIE and Boston University datasets [1] [2] [3]) we have also used web-cam captures in order to cover situations not present in the available datasets. Sample captures are available here : www.lifl.fr/˜dahmane/VIDEOS.

The paper is organized as follows. We first review the related work on head pose estimation in section 2. Then, we provide the methodology used for the estimation of the head pose using the symmetrical parts of the face in Section 3. In Section 4, we present the results of the evaluation process. The results of head pose estimation are discussed. Finally, we conclude and discuss the potential future work in Section 5.

2 Related work

In this section, we review the related work for head pose estimation regardless of the underlying descriptors and methodology. We analyse the existing methods in order to highlight advantages and disadvantages of each one. Then, we focus our attention on global approaches exploiting symmetry information. Even though the latter approaches are less popular, we strongly believe in the benefits of global symmetry for pose estimation.

Existing techniques for head pose estimation are summarised in [4] and can be categorized in six groups:

Model based approaches include geometric and flexible model approaches. Geometric approaches use the location of facial features such as eyes, mouth and nose and geometrically determine the pose from their relative configuration [5][6]. Flexible Model approaches use facial features to build a flexible model which fits to the image such that it conforms to the facial structure of each individual (AAM) [7]. However, accurate matching of a deformable face model to image sequences with large amounts of head movement is still a challenging task [8].

Classification-based approaches formulate the head pose estimation as a pattern classification problem. Several works have used a range of classifiers such as SVM [9][10]. Isarun and al. [11] uses random trees beside SVM. In [12] Kernel Principal Component Analysis (KPCA) is used to learn a non-linear subspace for each range of view. Then a test face is classified into one of the facial views using Kernel Support Vector Classifier (KSVC). Also, classification is achieved in [13] using a set of randomized ferns and in [14] Naive Bayes classifier are applied to estimate head pose.

Regression-based approaches consider pose angles as regression values. Several regressors are possible such as Convex Regularized Sparse Regression (CRSR) [15] and Gaussian Progress Regression (GPR) [16]. Murad and al. [17] proposed a method based on Partial Least Squares (PLS) Regression to estimate head pose. Support Vector Regressors (SVRs) are used to train Localized Gradient Orientation (LGO) histogram computed on detected facial region to estimate driver's head pose in [18]. Neural networks are one of the most used non-linear regression tools for head pose estimation. Tia and al. [19], use multiple cameras and estimate head pose by neural networks for each camera.

Template Matching approaches compare images or filtered images to a set of training examples and find the most similar. In [20] author represents faces with templates that cover different poses and for an input data uses correlation on model templates to achieve face recognition finding the best match. Similarity-to-prototypes philosophy is adopted by authors in [21] in order to calculate the pose similarity ratio.

Manifold Embedding approaches produce a low dimensional representation of the original facial features and then learn a mapping from the low dimensional manifold to the angles. Biased Manifold Embedding for supervised manifold learning is proposed in [22]. The incorporation of continuous pose angle information into one or more stage of the manifold learning process such as Neighbourhood Preserving Embedding (NPE) and Locality Preserving Projection (LPP) is studied in [23]. Dong [24] proposed Supervised Local Subspace Learning (SL2) to learn a local linear model where the mapping from the input data to the embedded space was learned using a Generalized Regression Neural Network (GRNN). In [25] author proposed the K-manifold clustering method, integrating manifold embedding and clustering.

Tracking approaches uses temporal information to improve the head pose estimation using the results of the head tracking [26]. In [11], a pedestrian tracker is applied to the heads video to infer head pose labels from walking direction and automatically aggregate ground truth head pose labels. Ba and al. [27] aims recognition of people's visual focus of attention by using a tracking system based on particle filtering techniques. KLT algorithm is used in [28] to track features over video frames in order to estimate 3D rotation matrix of the head.

Each approach has specific limitations. Appearance-based approaches suffer from information about identity and lighting which are contained in the face appearance. For template matching methods, the effect of identity can cause more dissimilarity then the pose itself. Most Manifold Embedding techniques have tendency to build manifold to identity as well as pose. Unlike model-based approaches are identity independent when the feature points used are linked to human morphology (anatomical points) and not to specific appearance points (mathematical points) like corners. Appearance-based approaches also require high computational time and this make it difficult to implement a real-time system. The model-based approaches are fast but sensitive to occlusion and usually require high resolution images. The difficulty lies in accurate detection of the facial features (morphological points) since all of the facial features are required (the outer corner of both eyes). High resolution imagery may not be available in many applications such as driver monitoring and e-learning systems. Also, model-based approaches [5], [7] require frontal view to initialize the system.

A specific family of approaches that exploit global features of the face, reducing dependency on identity and avoiding initialization of frontal pose, is represented by solutions that exploit facial symmetry. In [29], in order to detect possible regions for training a face in image, authors estimate the symmetry of the regions. The hypothesis is that the amount of symmetry can offer hints about head orientation.

2.1 Symmetry based approaches

The human perception of head pose is based upon two cues: the deviation of the head shape from bilateral symmetry, and the deviation of the nose orientation from the vertical [30]. Therefore, we presume that head pose is more related to the geometry of the face images and the symmetry of the face is a good indicator about the geometric configuration and the pose of the head. In the literature, there are a few symmetry based geometric head pose estimation methods.

Despite the fact that the human face is not perfectly symmetrical, facial symmetry of a person is significant and can be exploited. Some works dealing with the head pose estimation through feature points use the symmetrical property of the head. For instance, facial symmetry has been used as a visual intent indicator in [31] for people with disabilities. The face pose (roll and yaw angles) are estimated from a single uncalibrated view in [32] where the symmetric structure of human face is exploited by taking the mirror image of a test face image as a virtual second view. Facial feature points of the test and its mirror image are matched against each other in order to evaluate the pose. In [33] authors introduce gabor filters in order to enhance the symmetry computation process and estimate the yaw. Symmetry based illumination model proposed in [34] is based on three features (the two eyes and the nose tip). For every combination of two eyes and a nose, head pose is computed using a weak geometry projection and internally calibrated camera. In the context of face recognition, in [35] authors use the bilateral symmetry of the face to deduce if the pose is frontal or not. Beside intensity images, 3D data can be used for head pose estimation [36], [37].

Symmetry provides high-level knowledge about the geometry of face. We use the bilateral symmetry of the face to deal with the head pose estimation problem. We propose an approach to perform head pose estimation based on the symmetrical properties of the face.

3 Our approach

Our symmetry based approach aims at being non intrusive and do not require user collaboration. It has to be independent to user identity and can be deployed on still images as on videos. Our system use geometrical model which is not based on specific feature points. We propose an approach that joins the effectiveness of both local and global methods. We select symmetrical area on the face relative to skin pixels intensity and use the size of this area and it's orientation to estimate roll and yaw poses.

The proposed method (Figure 1) first detects the face using Viola Jones algorithm [38]. Preprocessing (histogram equalization) is applied in order to reduce illumination influence. Then the symmetry axis is searched in the area of the face. This task is performed using a symmetry detection algorithm. Once the head and the symmetry axis are detected, we extract the features of the symmetry. We deduce the roll angle from the orientation of the symmetry axis and estimate the yaw by analysing some characteristics of the symmetrical region. The method is described on figure 1. We can see that the symmetry detection allows the estimation of the roll angle and further the extraction of symmetric features.

In the following, we will bring out the correlation between symmetry and head pose by analysing the symmetrical regions of the face. We will detail the symmetry axis detection process and the characterization of the symmetry region. Then, we pass to the yaw estimation process by means of a Decision Tree Classifier.

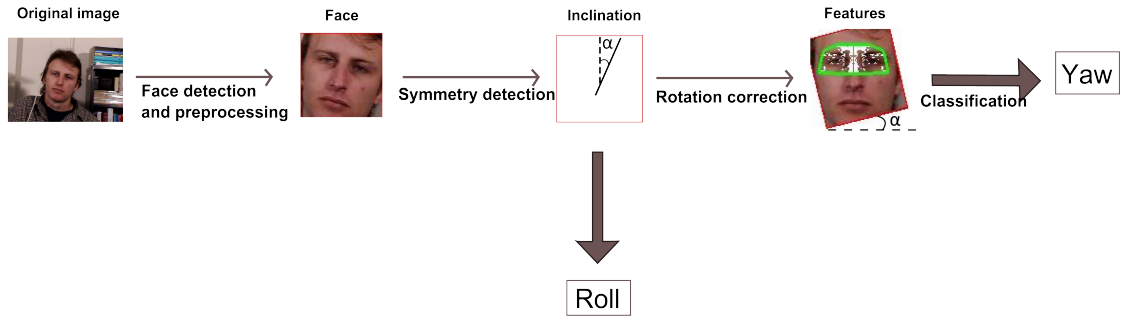

Fig. 1 Proposed approach.

3.1 Analysis of symmetrical regions on face

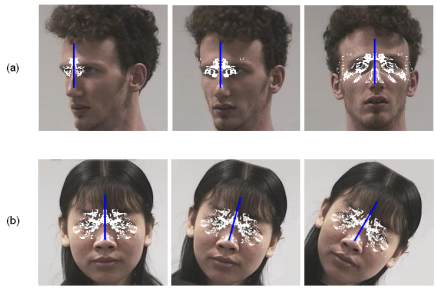

Fig. 2 (a) Variation in the size of the symmetrical region during yaw movement. (b) Variation in the angle of the symmetry axis during roll movement.

When the face is in front of the camera, the symmetry between its two parts appears clearly and the line which passes between the two eyes and nose tip defines the symmetry axis. However, when the head performs a motion, for example, a yaw motion, this symmetry decreases. We exploit the difference between the symmetries before and after the head rotation to deal with yaw movement.

Figure 2 shows the variation of the symmetrical region for various head yaw and roll poses. First, for the yaw movement, we analyse the amount of symmetrical parts under various yaw angles (Figure 2 (a)).

Let a and b two symmetrical points on the face. m is the middle of the segment $[ab]$. The projections of these points on the image plane are a_i, b_i, m_i. When the face is in front of the camera, the segments $[a_i m_i]$ and $[m_i b_i]$ are symmetrical with respect to m_i as shown in Figure 3 (a). When the head performs a yaw motion, the features points (a, b, m) are projected into (a'_i, b'_i, m'_i) (see Figure 3 (b)). Let ω' be the vanishing point associated to the direction of (a, b) in the image plane. Since the central

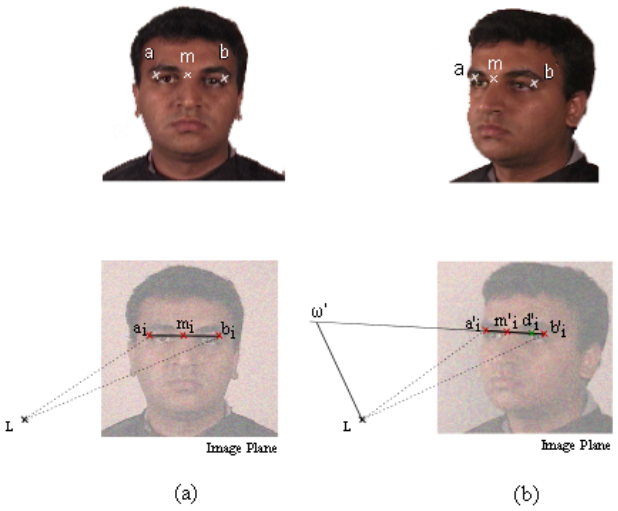

Fig. 3 (a) Projection of a segment (ab) when the face is in front of the camera, (b) Projection of the segment line after a yaw motion.

projection preserves the cross ratio [39], the cross-ratios of (a, b, m, ∞) and $(a'_i, b'_i, m'_i, \omega')$ are equal. We obtain:

$$\frac{ma}{mb} = \frac{m'_i a'_i}{m'_i b'_i} \div \frac{\omega' a'_i}{\omega' b'_i} \tag{1}$$

As the two members of the equation 1 are equal to one (as m is the middle of $[ab]$), the point m'_i is not the middle of $a'_i b'_i$ and its position depends on the position of $a'_i b'_i$ relatively to ω'. Since m is the symmetry centres of ab, the pixels of segment line $a'_i b'_i$ may satisfy a partial symmetry but in this case the symmetry center will not be the middle of $a'_i b'_i$. It will be m'_i and the symmetry will concerns the segments $m'_i a'_i$ and $m'_i d'_i$ where d'_i is located between m'_i and b'_i so as $m'_i a'_i = m'_i d'_i$ (see Figure 3 (b)).

Thus, after a yaw motion, the symmetry on the image plane is partial. The symmetrical part of a segment linking two symmetrical points on the face, is smaller

than the symmetrical part of the same segment before the movement.

Secondly, regarding the roll angle, we estimate that it corresponds to the angle of the symmetry axis (see Figure 2 (b)). We infer the pose angle from the inclination of the symmetry axis that we calculate in case of frontal view.

3.2 Symmetry axis detection

We use pixels intensity to detect symmetry in the image. Therefore, illumination influences the detection and in some cases, causes errors. We apply preprocessing on images before starting the symmetry detection in order to improve the robustness. We use the RGB space which gives more significant information about skin colour compared to grayscale and allows us to differentiate between face and background since one skin pixel is generally more similar to another skin pixel than to a background pixel. For this reason, we apply histogram equalization on each RGB color channels of the image in order to reduce illumination effect and then, we merge them back.

Our goal is to find the morphological symmetry of face, under different poses, provided that the desired symmetry does not disappear completely from the image (e.g. when yaw angle exceeds 45°). Our algorithm is based on Stentiford [40]. It was necessary to adapt the initial algorithm that highlights the symmetries present in the image regardless of what the image represents, a face or an object. After detecting the face, we consider an ellipse inside and we set our region of interest to be the top half of the ellipse. This part of the face is chosen because upper part is more affected by head rotations. The change in the size of the symmetric region after a right/left rotation is greater in the region of the eyes than that of the mouth.

In order to detect the position P and the orientation α of the image symmetry axis, we vary α from α_{min} to α_{max} with a step α_{step}. Then, for each inclination, we seek the position of the symmetry axis. Once all the inclinations tested, a vote is performed considering as best axis the one which accounts the greatest number of symmetrical pixels as well as being the closest to the face centre. We consider the distribution of the symmetry axes $A_{i\{P_i,\alpha_i\}}$, each of them weighted by the number of the local symmetries which it satisfies. We take the n maximum of this distribution and vote for the axis $A_{\{P,\alpha\}}$ which minimize the distance to the face centre C such that:

$$d(C, A_{\{P,\alpha\}}) = min\{d(C, A_{i\{P_i,\alpha\}})\} i \in [1, n] \quad (2)$$

Detect the symmetry relative to an inclination:

The region of interest of the image is divided into small overlapping square blocks "cells" with side s. We search for local symmetries via the image cells by searching the symmetrical cell of each non-homogeneous cell in the region of interest. When we find two symmetrical cells, we can determine the position of their symmetry axis. This local symmetry axis passes perpendicularly in the middle of the strip which passes through the two cells. After we detect all the symmetry axes $A_{i\{P_i,\alpha\}}$ with i ranges from 1 to the number of axis positions for a given inclination α. We vote for the best axis $A_{\{P,\alpha\}}$ using the same mechanism as defined previously.

Define the local symmetries:
We test for a match between the original cell and all its mirror cells relative to α (see Figure 4) until a match is found. The location of each mirror cell is calculated with the equation 3. The coordinates of a pixel (x,y) in the reflected position are (x_i,y_i). We vary x_i along the width of the region of interest and obtain y_i.

$$y_i = y + ((\tan\alpha) \times (x_i - x)) \quad (3)$$

If two cells match, we consider them symmetric. Mirror cells lie on the strip which passes through the original cell and that is inclined by an angle $\alpha + \pi/2$.

- Two cells match if each pixel on the diagonal of the original cell matches its corresponding pixel on the mirror cell.
- A pixel matches another one if the intensity difference of the three channels does not exceed a given threshold ε.

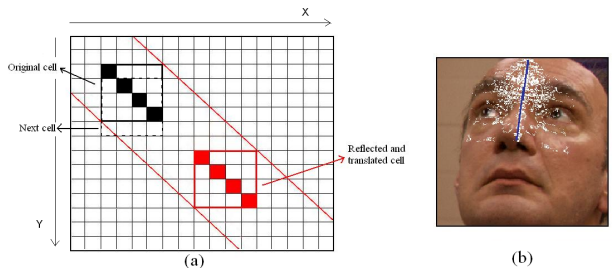

Fig. 4 Symmetry axis detection.

Pseudocode for symmetry detection:
ROI: The region of interest (inside the ellipse)
C: A cell belonging to the ROI
r: Diagonal of the cell C
C^r: Mirror cell of C (after reflection)

x^r: First pixel belonging to the diagonal of C^r
Correspondence: Boolean to indicate the correspondence between two cells.

```
α ← α_min
while α < α_max do
    while ROI do
        Take a cell C
        if C non-homogeneous then
            x^r ← widthROI
            Correspondence ← false
            while Correspondence = false AND
            x^r > r do
                Define C^r (equation 3)
                if C^r ∈ ROI then
                    Test correspondence between C
                    and C^r
                    if correspondence then
                        Save A_{i{P_i,α,nbrSym}}
                    end
                end
                x^r ← x^r − 1
            end
            Take n maximum in the distribution of
            A_{i{P_i,α,nbrSym}}
            Vote for A_{P,α,nbrSym} via equation 2
            Save A_{P,α,nbrSym}
        end
    end
    α ← α + α_ε
end
Take n maximum in the distribution of
A_{P,α,nbrSym}
Vote for A via equation 2
```

Algorithm 1: Symmetry axis detection

The interval and the step of α influence the results. A small step gives more accuracy but takes more calculation time and requires a large amount of storage. We set the step according to the interval so that we do not obtain a high-dimensional distribution. Also, a big interval may provide symmetries which do not correspond to the bilateral symmetry searched. To this, we set α_{max} to not exceed 135° and α_{min} not under 45° because the natural movement of the roll does not exceed 45° on each side.

One can see results of symmetry axis detection on Figure 2. We set $s = 20$ and $\varepsilon = 25$, $n = 3$, $\alpha_{min} = 80°$, $\alpha_{max} = 100°$ and $\alpha_{step} = 3°$.

3.3 Features

Once we have detected symmetry axis, and rotated the image with respect to the axis inclination, we extract symmetrical features. We test for a match between all the pixels and their symmetrical one related to the detected symmetry axis. This time, differently from the previous step (symmetry axis detection), pixels should not be part of a cell. The pixels are tested one by one, in order to define the region of symmetry without excluding homogeneous area (see Figure 5). In this way, detection of the symmetrical region is not sensitive to pixel matching process since we use all the texture. We can find x_2, the symmetrical pixel of x_1, with the equation 3. If the difference in intensity between the two pixels is greater than a certain threshold, we decide that the two pixels are not symmetric. Then, we use the convex hull encompassing symmetrical pixels to characterise the geometric features: the size of the symmetrical region (as shown in Figure 5).

After experimental attribute selection, vertical measurement are not kept because not useful for yaw movement. Therefore, as symmetrical features, we use the width of the hull which contains symmetrical pixels and the mean distance of all symmetrical pixels to the axis of symmetry. We define the width as euclidean distance between the two most distant pixels.

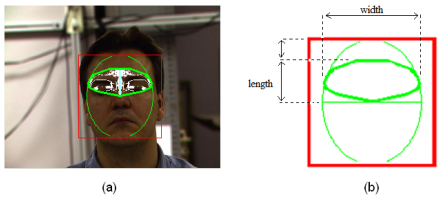

Fig. 5 Examples of extracting features. (a) Computing a convex hull which includes the symmetrical pixels. (b) Measures relating to the symmetrical region.

3.4 Yaw estimation

3.4.1 Decision tree classifier

In order to determine the amount of yaw motion, a Decision Tree classifier is trained using the relative features (width of the symmetrical region and the mean distance of symmetrical pixels) extracted from the symmetrical parts according to the amount of yaw motion. Each class of the classifier corresponds to a discrete pose. To increase performance of prediction, we use the Alternating Decision Tree which is based on boosting [41]. The tree alternates between prediction nodes and decision nodes. The root node is a prediction node and contains a value for each class. The prediction values are used as a measure of confidence in the prediction.

The set of head pose images used for learning represents the angles for which the symmetry axis is properly detected. The poses are discrete and vary from -45° (left) to +45° (right).

We start by extracting features from the region of interest as described in section 3.3. Then, we construct the model from the feature vectors derived from images of several people recorded in different poses. Right and left poses with the same angle are gathered in the same class as they contain the same amount of symmetry and, therefore, the same information. Thus, to estimate $2 * n + 1$ discrete poses (n lateral right poses, n lateral left poses and 1 frontal pose), the classifier has $n + 1$ classes. For this, we use $2 * n + 1$ images per subject to represent the $2 * n + 1$ poses.

The root contains null values as prediction for the $n+1$ classes. The first level contain decision nodes based on the values of the feature vector attributes, followed by prediction nodes for each class and so on until the leaves. The sum of the prediction values crossed when following all paths for which all decision node are true, is used to classify a given instance. The class which has the biggest prediction value is the predicted class.

As we use a supervised classification approach, we first have to train the alternating decision tree classifier using the same number of images per person as the number of classes. With the constructed tree, we can predict the yaw for various test face images. Training and testing images do not have to be from the same dataset.

3.4.2 Left vs Right poses

To differentiate between left and right poses, we use the difference in intensity between the skin and the background. Our assumption is that a pixel on the face is more similar to another pixel on the face than to a pixel on the background.

We take a pixel located on the symmetry axis to ensure that it is on the face. We compute the average intensity of the pixels surrounding and consider this value as a reference. If the symmetry axis is closer to the left contour (resp. right contour) of the face, then the face is oriented to the left (resp. right). We calculate two values : the difference between the reference value and the average intensity of pixels on the left side and the same difference for the reference value and the right side of the axis. If the difference is bigger on the left side (resp. right), we conclude that the face on the image is oriented to the left (resp. right).

With this method, we determine which side the pose is oriented and this information is combined with the degree of orientation estimated by the Decision Tree in order to obtain the yaw head pose.

4 Experimental results and discussion

We evaluate the obtained model in order to validate the features extracted from symmetry. We first evaluate the approach using the Face Pix [1] dataset which is ideal for the yaw motion. It consists of poses in the interval $\pm 90°$ at $1°$ increments. This allow us to form several class configurations as explained in section (3.4.1) (e.g. 10 classes for 19 poses). Also, we test our approach on the CMU PIE dataset [2] which gives more variability in term of illumination and expressions (e.g. eyes closed or smile). In addition to image datasets, we test the video sequences of the Boston University (BU) dataset [3]. In BU dataset, subjects are doing free movements including yaw and roll variations. This allow us to estimate the roll (in-plane rotation) accuracy besides the yaw. Poses in the videos are predicted using the model built with the Face Pix dataset. Video sequences are also recorded in the lab, to reproduce situations of partial occlusion not present in the available datasets. In all experiments, we use the same parameters $\varepsilon = 25$, $s = 20$ et $n = 3$ for symmetry axis detection. The interval of α is $[85°, 95°]$ for FacePix and CMU PIE and $[45°, 135°]$ for BU dataset with a $step = 3°$. The results of our experiments are presented below.

4.1 Face Pix dataset

We use the Face Pix dataset [1] to build a head pose model and to evaluate it. The FacePix database consists of three sets of face images : variable pose, variable dark illumination and variable light illumination. The sets of variable illumination images have only frontal pose. This is why we use only the set of variable poses which is composed of 181 pose images of 30 different subjects. Among the 181 poses, we use poses varying from $-45°$ to $+45°$ because when exceeding this interval, the bilateral symmetry disappears from the image plane.

We test several configurations, changing the number of classes each time. Figure 6 shows the confusion matrix for three classifiers : 19 discrete poses associated to the yaw angles from -45° to 45° with 5° step (10 classes), 9 discrete poses associated to the yaw angles from -40° to 40° with 10° step (5 classes) and 7 discrete poses associated to the yaw angles from -45° to 45° with 15° step (4 classes). One can see that the estimated pose is in the diagonal of the matrix. However, the 7 poses model had the higher classification rate but further experiments on continuous image sequences (the BU dataset's videos) reported in section (4.3) show that the model of 19 poses give more accuracy on this dataset.

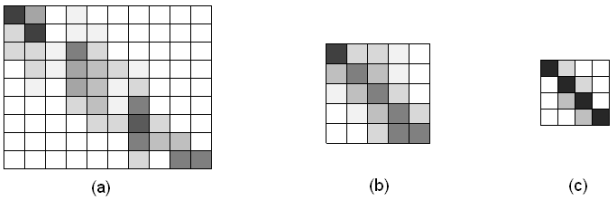

Fig. 6 Confusion matrix associated to: (a)19 poses classifier with 5° step. (b)9 poses classifier with 10° step. (c)7 poses classifier with 15° step.

In order to evaluate the model, we split the data into 6 equal subsets and performed 6-fold cross validation. In each run, 5 subsets are used as the training set and the rest is used as a test set. The subjects in the training and test set are completely distinct since each subject is taken only once. On this dataset, we test the sensitivity of the method to the symmetry axis detection accuracy. We annotated the position of the head and the position and the orientation of the symmetry axis in order to compare results in a semi-automatic and a fully-automatic settings.

A detailed description of the results with 7 poses is shown in Table 1.

Table 1 Classification rates and Mean Absolute Errors (MAE) for FacePix dataset in semi and fully automatic modes.

Data	Accuracy (%)	MAE (°)
Head and symmetry axis annotated	82.38	2.71
Head annotated and symmetry automatic	81.90	2.78
Head and symmetry detection automatic	79.63	3.14

When removing errors related to head detection and/or symmetry axis detection, the results outperform those of the completely automatic mode. The latest one are not much worse, since the classification accuracy reaches 79.6% for the seven poses model.

4.2 CMU PIE dataset

The CMU Pose, Illumination and Expression dataset [2] contains images of 68 subjects with a step of 22.5° between poses. In our experiments we use the image set corresponding to variable expression (4 different expressions), the one recorded under variable lighting conditions (21 different flash orientations) and the set with subjects talking. Concerning the first set (Expression), we use images of poses between 45° and −45°. We built a classifier for each set to study apart the robustness to varying expressions and lighting. We calculate the classification rate for each classifier using 6-folds cross-validation. We also merge all images in one encompassing set. The challenge with this dataset is the variable lighting set. In this case, when there is an intense light source in a lateral side, the scene loses it's symmetry.

Table 2 shows results for each set and those considering all images in one encompassing set. To achieve illumination invariance, the RGB histogram equalization is not sufficient. We apply a discrete cosine transform (DCT) based normalization technique [42] to the full image. A number of DCT coefficients are truncated to minimize illumination variations since the variations mainly lie in the low frequency band. This truncation affect the matching process in the Expression set. The accuracy drop from 72.57% to 49.81%. Unlike Talking and Lighting sets where DCT normalization did more good than bad. The illumination affects strongly the matching of symmetries than the noise added by the normalization. On the other hand, DCT normalization gives better results on sets with a great number of learning images. In the CMU Expression set, for each person, each pose is represented by 3 or 4 images (neutral, blinked, smiling and for certain subject with glasses). However, in the Talking set, each pose has 60 images and in the Light set, 23 images are recorded for each pose.The large number of images used for learning offset the loss due to the normalization and even allowed slightly improve the accuracy for the Talking set.

Table 2 Results for the CMU PIE dataset.

Data	Classification accuracy (%)	
	RGB Equalization	DCT
CMU Expression	72.57	49.81
CMU Talking	81.04	87.63
CMU Lighting	72.51	85.90
CMU PIE	72.48	82.26

4.3 Videos

We also test on videos as we aim to use the solution in real environment for having real pose related feedback. We test our method on the video sequences of the Boston University head pose dataset [3]. We recall that the inclination of the symmetry axis corresponds to the roll angle in case of frontal view. The yaw and the roll are calculated over all the frames in order to compare with ground truth. In the experiments, we have used the alternating decision tree trained on FacePix dataset as it covers better the range of face poses than CMU PIE which has widely spaced poses (22.5° between poses). We ensure that the size of the face in the BU images is the same than in FacePix dataset. The best results for the yaw are obtained using the model built with 19 discrete poses and 5° step from the FacePix dataset, giving 5.24° mean absolute error (MAE), 6.80° root mean squared error (RMSE) and a standard deviation (STD)

of 4.33°. Results for the yaw and the roll are shown in Table 3.

Table 3 Results for the BU dataset.

	RMSE (°)	MAE (°)	STD (°)
Roll	4.39	2.57	3.56
Yaw (FacePix model - 5° intervall)	7.60	5.12	5.62
Yaw (FacePix model - 15° intervall)	6.80	5.24	4.33

We exploit the temporal information contained in the video stream in order to reduce calculation time. We use the position and the orientation of the symmetry axis of a given frame to reduce the search interval in the next frame. We perform a check in every 10 frames (approximately one second).

4.4 Resolution and occlusion

We have conducted experiments which indicate that the facial symmetry is a good geometrical indicator for head pose when the local geometric features (such as eyes, nose or mouth) are missed due to occlusions or wrong detections. When the head rotation exceeds 30° (in a left/right rotation), some feature points disappear from the image plane but partial symmetry still exists.

In order to measure the robustness of the approach, we generate low resolution images from the FacePix dataset where the head size were 80×80. We resize the head to generate two head image sets, the first is 40×40 pixels and the second 25×25 pixels. We succeeded in detecting the symmetric features, thing that can't be done when relying on specific feature points. We built a 9 pose classifier for both sets using the parameters $\varepsilon = 25$, $s = 2$ et $n = 3$, $85° <= \alpha <= 95°$. The accuracy of the first classifier is 74.1% and that of the second is 63.8%. We can see that the accuracy drop from 79.6% because the method is based on local symmetries and our algorithm is sensitive to symmetry axis calculation. On very low resolution images, the local symmetries are not enough relevant. But results are not very bad for heads which are 25×25 pixels.

We also test the system with web-cam in the laboratory simulating local partial occlusions. As the process does not need interest points, partially occluded faces can be processed since there is at least one couple of symmetrical pixels on the image. To do so, all the texture pixels in the region of interest, contribute to the demarcation of the symmetrical area. This can be seen in figure 7.

Fig. 7 Sample frames from video sequences taken in lab.

4.5 Summary and comparison with the state of the art

The main advantage of the method is that the calculation can start at any pose, without any initialisation, since the head and the symmetry axis are automatically detected for poses between -45° and +45°. Also, new face images can be classified easily meaning a built model.

In video sequences where the head is performing free movements, wrong detections often occur. To resolve this issue, we exploit the continuity of movement. We exclude detections which are very far from the 3 previous frames considering them as wrong. We use instead an interpolated position of the head. The process is then, fully automatic but sensitive to the accuracy of head detection and symmetry axis calculation. The system is robust to changes in lighting condition, expression and also to identity informations since the method is geometric. Besides, no specific points are needed to be detected on the face. So, closed eyes or partially occluded face give the same results as complete face.

We compare our results with others which used the same datasets. Tian et al. [19] obtained ?? % of good classification on CMU PIE dataset and 82% for us. Tables 4 and 5 shows results on FacePix and BU datasets expressed in MAE, RMSE, STD and classification accuracy (Acc). From these results, it is shown that our method provide comparable results on CMU PIE and BU datasets. On FacePix, manifold embedding methods give good results but there is no explicit solution for out-of-sample embedding in an LLE and LE manifold [4]. These methods are not automatic unlike ours. New data can be classified easily through a model of examples already built.

5 Conclusion

We presented a new approach to perform head pose estimation. We exploit bilateral symmetry of the face to deal with roll and yaw motions. The orientation of

Table 4 Comparison of the yaw results with the state of the art using FacePix dataset.

Method	resolution	MAE (°)	Acc (%)
Hao et al. 2011 (Regression)	60 ×60	6.1	-
Xiangyang et al. 2010*(K-manifold clustering)	16 ×16	3.16	-
Vineeth et al. 2007*(Biased Isomap)	32 ×32	5.02	-
Vineeth et al. 2007*(Biased LLE)	32 ×32	2.11	-
Vineeth et al. 2007*(Biased LE)	32 ×32	1.44	-
Proposed	80 ×80	3.14	79.63
	40 x 40		
	25 x 25		

* A significant drawback of manifold learning techniques is the lack of a projection matrix to treat new data points.

Table 5 Comparison of the BU dataset results with the state of the art.

		RMSE (°)	MAE (°)	STD (°)
Valenti et al. 2012	Yaw	6.10[a]	-	5.79[a]
	Roll	3.00[a]	-	2.82[a]
Morency et al. 2010	Yaw	-	4.97	-
	Roll	-	2.91	-
Proposed	Yaw	6.80	5.24	4.33
	Roll	4.39[b]	2.57[b]	3.56[b]

[a] Eye cues used, the pose is estimated only when eyes are detected.
[b] The roll is estimated in case of frontal view.

the symmetry axis indicates the roll angle of the head. The symmetrical region of the face with respect to this orientation provides us features such as the width of region which allow us to classify and then, to predict yaw angles. Symmetrical features may be extracted without the detection of special facial landmarks and no calibration nor initial frontal pose are required. The results obtained by our approach have been evaluated using public datasets and they outline the good performance of our algorithm with regard of the state of the art methods. In our future work, we will nominate new features which allow us to estimate combined yaw and pitch pose. We will also explore more complicated regression methods to achieve the two degrees of freedom. we are planning also to explore temporal correlation obtained from the head tracking to extend the range of motion.

Acknowledgements This work was conducted in the context of the ITEA2 "Empathic Products" project, ITEA2 1105, and is supported by funding from DGCIS, France.

References

1. J. Black, M. Gargesha, K. Kahol, P. Kuchi, and S. Panchanathan, "A framework for performance evaluation of face recognition algorithms," in *ITCOM, Internet Multimedia Systems II, Boston*, 2002.
2. T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, 2003.
3. R. Valenti and T. Gevers, "Robustifying eye center localization by head pose cues." in *IEEE conference on Computer Vision and Pattern Recognition*, 2009.
4. E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 4, pp. 607–626, 2009.
5. J.-G. Wang and E. Sung, "Em enhancement of 3d head pose estimated by point at infinity," *Image Vision Comput.*, vol. 25, no. 12, pp. 1864–1874, Dec. 2007.
6. Y. Pan, H. Zhu, and R. Ji, *3-D Head Pose Estimation for Monocular Image*, ser. Fuzzy Systems and Knowledge Discovery. Springer, 2005.
7. S. Baker, I. Matthews, J. Xiao, R. Gross, T. Kanade, and T. Ishikawa, "Real-time non-rigid driver head tracking for driver mental state estimation," in *in 11th World Congress on Intelligent Transportation Systems*, 2004.
8. T. C. Angela Caunce, Chris Taylor, "Improved 3d model search for facial feature location and pose estimation in 2d images," *BMVC*, 2010.
9. J. Huang, X. Shao, and H. Wechsler, "Face pose discrimination using support vector machines (svm)," in *International Conference on Pattern Recognition (ICPR)*, 1998.
10. M. Dahmane and J. Meunier, "Object representation based on gabor wave vector binning : An application to human head pose detection," *ICCV*, 2011.
11. D. S. T. S. T. O. Y. S. I. Chamveha, Y. Sugano and A. Sugimoto, "Appearance-based head pose estimation with scene-specific adaptation," *ICCV*, 2011.
12. S. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang, "Kernel machine based learning for multi-view face detection and pose estimation," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, 2001, pp. 674 –679 vol.2.
13. B. Benfold and I. Reid, "Colour invariant head pose classification in low resolution video," *BMVC*, 2008.
14. Z. Zhang, Y. Hu, M. Liu, and T. Huang, "Head pose estimation in seminar room using multi view face detectors," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science, R. Stiefelhagen and J. Garofolo, Eds. Springer Berlin Heidelberg, 2007, vol. 4122, pp. 299–304.
15. F. S. Z. S. Y. T. Hao Ji, Risheng Liu, "Robust head pose estimation via convex regularized sparse regression," *ICIP*, 2011.
16. A. Ranganathan and M.-H. Yang, "Online sparse matrix gaussian process regression and vision applications," *ECCV*, 2008.
17. J. G. Murad Al Haj and L. S. Davis, "On partial least squares in head pose estimation: How to simultaneously deal with misalignment," *CVPR*, 2012.
18. E. Murphy-Chutorian, A. Doshi, and M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in *Intelligent Transportation Systems Conference, ITSC*. IEEE, 2007, pp. 709–714.
19. Y. li Tian, L. Brown, J. Connell, S. Pankanti, A. Hampapur, A. Senior, and R. Bolle, "Absolute head pose estimation from overhead wide-angle cameras," in *In IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.
20. D. J. Beymer, "Face recognition under varying pose," *CVPR*, pp. 756–761, 1994.
21. S. G. Jamie Sherrah and E.-J. Ong, "Understanding pose discrimination in similarity space," *BMVC*, 1999.
22. J. Y. Vineeth Nallure Balasubramanian and S. Panchanathan, "Biased manifold embedding: A framework for person-independent head pose estimation," *CVPR*, 2007.

23. C. BenAbdelkader, "Robust head pose estimation using supervised manifold learning," *ECCV*, 2010.
24. F. D. l. T. H. B. Dong Huang, Markus Storer, "Supervised local subspace learning for continuous head pose estimation," *CVPR*, 2011.
25. W. L. Xiangyang Liu, Hongtao Lu, "Multi-manifold modeling for head pose estimation," *ICIP*, 2010.
26. R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, 2012.
27. S. O. Ba and J.-M. Odobez, "Multiperson visual focus of attention from head pose and meeting contextual cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 101–116, 2011.
28. M. Nabati and A. Behrad, "3d head pose estimation and camera mouse implementation using a monocular video camera," *Signal, Image and Video Processing*, pp. 1–6, 2012.
29. H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR '98, 1998, pp. 38–.
30. H. R. Wilson, F.Wilkinson, L. Lin, and M. Castillo, "Perception of head orientation," *Vision Research*, vol. 40, no. 5, pp. 459–472, 2000.
31. T. Luhandjula, E. Monacelli, Y. Hamam, B. van Wyk, and Q. Williams, "Visual intention detection for wheelchair motion," in *International Symposium on Visual Computing (ISVC)*, 2009, pp. 407–416.
32. S. D. Vinod Pathangay and T. Greiner, "Symmetry-based face pose estimation from a single uncalibrated view," *8th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2008),The Netherlands*, pp. 1–8, 2008.
33. B. Ma, A. Li, X. Chai, and S. Shan, "Head yaw estimation via symmetry of regions," 2013, pp. 1–6.
34. M. Gruendig and O. Hellwich, "3d head pose estimation with symmetry based illumination model in low resolution video," in *Lecture Notes in Computer Science*, vol. 3175. Springer, 2004, pp. 45–53.
35. J. Harguess, S. Gupta, and J. Aggarwal, "3d face recognition with the average-half-face," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.
36. K. Hattori, S. Matsumori, and Y. Sato, "Estimating pose of human face based on symmetry plane using range and intensity images," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2, 1998, pp. 1183–1187 vol.2.
37. Z. Gui and C. Zhang, "3d head pose estimation using non-rigid structure-from-motion and point correspondence," *IEEE TENCON*, 2006.
38. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511 – I–518 vol.1.
39. H. Coxeter, *Projective Geometry*, ser. Fuzzy Systems and Knowledge Discovery. Springer-Verlag 2nd Revised edition, 2003.
40. F. Stentiford, "Attention based facial symmetry detection," in *In Proc. ICAPR 2005*, 2005.
41. G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall, "Multiclass alternating decision trees," in *ECML*. Springer, 2001, pp. 161–172.
42. W. Chen, M. J. Er, and S. Wu, "Illumination Compensation and Normalization for Robust Face Recognition Using Discrete Cosine Transform in Logarithm Domain," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 36, pp. 458–466, 2006.